

Algoritma dan Pemrograman R

Bakti Siregar, M.Sc

20 Agustus 2022

Contents

Kata Pengantar	5
Ringkasan Pembelajaran	5
Tim Penyusun	6
Ucapan Terima Kasih	7
Masukan & Saran	7
1 Pengenalan R?	9
1.1 Fitur Utama R	9
1.2 Mengapa Belajar R?	10
1.3 Download R & Rstudio:	10
1.4 Tutorial Instal R & Studio	10
1.5 Video Instalasi R & RStudio	12
1.6 Interface R & RStudio:	12
1.7 Sintaks Dasar R	13
1.8 Bantuan (Help) R	13
1.9 Shortcut Penggunaan Rstudio	14
1.10 Praktikum	14
2 Dasar Pemrograman R	15
2.1 Variabel	15
2.2 Operator	16
2.3 Tipe Data	19
2.4 Statistika Dasar	20

2.5	Contoh Pemrograman Dasar	20
2.6	Latihan	22
3	Struktur Data	25
3.1	Vektor	26
3.2	Matriks	27
3.3	Array	28
3.4	Faktor	28
3.5	Data Frame	28
3.6	Lists	29
3.7	Rekayasa Data Frame	31
3.8	Latihan	34
4	Referensi	35

Kata Pengantar

Bahasa pemrograman R telah menjadi alat yang kuat bagi para ilmuwan data, analis statistik, dan praktisi analisis numerik di seluruh dunia. Dengan kemampuan yang luar biasa dalam manipulasi data, visualisasi, dan analisis statistik, R memungkinkan para profesional untuk menggali wawasan berharga dari kumpulan data yang kompleks.

Dalam buku ini, penulis menyediakan materi dasar-dasar bahasa pemrograman R hingga tingkat yang lebih mendalam. Penulis juga menjelaskan beberapa konsep-konsep penting, sintaksis dasar, struktur data, serta memberikan contoh nyata tentang bagaimana R dapat digunakan dalam berbagai konteks. Modul ini dirancang untuk membantu pembaca yang baru mengenal pemrograman maupun yang telah memiliki pengalaman sebelumnya dalam bahasa lain.

Ringkasan Pembelajaran

Adapun isi pembelajaran dalam modul ini adalah sebagai berikut:

- **Minggu 1-2: Dasar Pemrograman R**
 - Pengenalan konsep dasar pemrograman.
 - Pengenalan lingkungan dan pengaturan awal bahasa R.
 - Menulis program sederhana dalam R.
- **Minggu 3-4: Variabel dan Tipe Data**
 - Mengenal tipe data dasar dalam R: numerik, karakter, logika.
 - Deklarasi dan penggunaan variabel.
 - Konversi antar tipe data.
- **Minggu 5-6: Struktur Kontrol**
 - Penggunaan pernyataan kondisional (if, else).
 - Penggunaan perulangan (for, while) untuk mengatur alur program.
 - Studi kasus penggunaan struktur kontrol dalam pemrograman.
- **Minggu 7-8: Fungsi**

- Pengenalan konsep fungsi dalam pemrograman.
 - Membuat dan memanggil fungsi dalam R.
 - Penggunaan parameter dalam fungsi.
- **Minggu 9-10: Struktur Data**
 - Pengenalan array dan matriks dalam R.
 - Penggunaan vektor dan faktor.
 - Pengenalan konsep dataframe untuk manipulasi data tabular.
- **Minggu 11-12: Algoritma Dasar**
 - Pengenalan konsep algoritma dan kompleksitas.
 - Pemahaman tentang pencarian dan pengurutan.
 - Implementasi algoritma pencarian dan pengurutan sederhana dalam R.
- **Minggu 13-14: Pengenalan Analisis Data**
 - Pengenalan pustaka dasar untuk analisis data di R.
 - Pemahaman tentang statistik dasar dan visualisasi data.
 - Penerapan analisis sederhana pada dataset kecil.
- **Minggu 15-16: Proyek Akhir** Siswa diminta untuk membuat proyek kecil menggunakan R yang menggabungkan konsep-konsep yang telah dipelajari. Proyek dapat berupa analisis data, pemecahan masalah, atau aplikasi sederhana.

Tim Penyusun

Berikut ini adalah nama dan biografi singkat para penulis:

- **Bakti Siregar, M.Sc** adalah Ketua Program Studi di Jurusan Statistika Universitas Matana. Lulusan Magister Matematika Terapan dari National Sun Yat Sen University, Taiwan. Beliau juga merupakan dosen dan konsultan Data Scientist di perusahaan-perusahaan ternama seperti JNE, Samora Group, Pertamina, dan lainnya. Beliau memiliki antusiasme khusus dalam mengajar Big Data Analytics, Machine Learning, Optimisasi, dan Analisis Time Series di bidang keuangan dan investasi. Keahliannya juga terlihat dalam penggunaan bahasa pemrograman Statistik seperti R Studio dan Python. Beliau mengaplikasikan sistem basis data MySQL/NoSQL dalam pembelajaran manajemen data, serta mahir dalam menggunakan tools Big Data seperti Spark dan Hadoop. Beberapa project beliau dapat dilihat di link berikut: Rpubs, Github, Website, dan Kaggle.

Ucapan Terima Kasih

Kami berharap modul ini akan menjadi panduan yang bermanfaat bagi Anda dalam menguasai bahasa pemrograman R. Semoga dengan memahami konsep-konsep yang disajikan dalam modul ini, Anda akan dapat mengaplikasikan R dalam proyek-proyek analisis data dan statistik yang sebenarnya.

Terima kasih kepada semua yang telah berkontribusi dalam pembuatan modul ini, serta kepada Anda, pembaca, yang telah memilih modul ini sebagai sumber pengetahuan Anda. Kami berharap Anda menikmati perjalanan Anda dalam memahami bahasa pemrograman R.

Masukan & Saran

Semua masukan dan tanggapan Anda sangat berarti bagi kami untuk memperbaiki modul ini kedepannya. Bagi para pembaca/pengguna yang ingin menyampaikan masukan dan tanggapan, dipersilahkan melalui kontak dibawah ini!

Email: dsciencelabs@outlook.com

Chapter 1

Pengenalan R?

R adalah bahasa pemrograman dan lingkungan komputasi yang digunakan untuk analisis statistik, visualisasi data, pengolahan data, dan pemodelan prediktif. R dikembangkan oleh Ross Ihaka dan Robert Gentleman di Universitas Auckland, Selandia Baru. R menjadi populer dalam dunia analisis data dan ilmu data karena kemampuannya dalam mengolah dan menganalisis data secara efisien.

1.1 Fitur Utama R

1. **Open Source:** R adalah perangkat lunak open source yang dapat diunduh dan digunakan secara gratis.
2. **Fleksibilitas:** Anda dapat membuat fungsi sendiri, mengontrol alur program, dan mengakses berbagai pustaka eksternal.
3. **Mengimpor dan Mengekspor Data:** R mendukung berbagai format file, seperti CSV, Excel, SQL, dan format data lainnya.
4. **Data Manipulasi:** R memiliki pustaka seperti dplyr dan tidyr yang memudahkan manipulasi dan transformasi data.
5. **Lingkungan Komputasi:** R tidak hanya bahasa pemrograman, tetapi juga lingkungan komputasi lengkap yang menyediakan alat untuk analisis dan visualisasi data.
6. **Statistik dan Analisis Data:** R memiliki beragam pustaka dan paket yang mendukung analisis statistik, visualisasi data, dan pemodelan prediktif.
7. **Grafik dan Visualisasi:** R memiliki kemampuan visualisasi yang kuat dengan pustaka seperti ggplot2 untuk membuat grafik yang informatif dan menarik.
8. **Komunitas Aktif:** Komunitas R sangat aktif, dan ada banyak sumber

daya online, forum, dan pustaka yang dapat membantu dalam pembelajaran dan pemecahan masalah.

1.2 Mengapa Belajar R?

Berikut ini adalah beberapa alasan mengapa penting untuk belajar R:

1. **Pengolahan Data:** R dapat membantu Anda membersihkan, merubah format, dan mengolah data sebelum analisis lebih lanjut.
2. **Analisis Data:** R adalah alat yang kuat untuk menganalisis data, membuat visualisasi yang menarik, dan mengidentifikasi pola dalam dataset.
3. **Karir di Ilmu Data:** Penguasaan R menjadi salah satu keahlian yang sangat dihargai dalam industri ilmu data dan analisis data.
4. **Komunitas Besar:** Anda akan menjadi bagian dari komunitas besar yang mendukung dan berkontribusi dalam pengembangan R serta membagikan pengetahuan.

1.3 Download R & Rstudio:

1. **Unduh dan Instalasi:** Kunjungi situs resmi R (<https://www.r-project.org/>) untuk mengunduh installer sesuai dengan sistem operasi Anda.
2. **RStudio (Opsional tapi Disarankan):** RStudio adalah lingkungan pengembangan terintegrasi (IDE) yang mempermudah pengembangan dalam R. Anda dapat mengunduh RStudio (<https://www.rstudio.com/>) dan menggunakannya untuk menulis dan menjalankan kode R.

1.4 Tutorial Instal R & Studio

Berikut adalah panduan langkah demi langkah untuk menginstal R dan RStudio

1.4.1 Instalasi R

R adalah bahasa pemrograman inti yang digunakan oleh RStudio. Ikuti langkah-langkah di bawah ini untuk menginstal R:

Windows

1. Kunjungi situs resmi R di <https://cran.r-project.org/mirrors.html>.
2. Pilih cermin (mirror) terdekat untuk mengunduh installer R.

3. Unduh installer R untuk Windows dan jalankan file installer yang diunduh.
4. Ikuti panduan instalasi, pilih opsi default kecuali jika Anda tahu persis apa yang Anda lakukan.
5. Setelah instalasi selesai, R akan terinstal di komputer Anda.

MacOS

1. Kunjungi situs resmi R di <https://cran.r-project.org/mirrors.html>.
2. Pilih cermin (mirror) terdekat untuk mengunduh installer R.
3. Unduh installer R untuk macOS dan jalankan file installer yang diunduh.
4. Ikuti panduan instalasi, pilih opsi default kecuali jika Anda tahu persis apa yang Anda lakukan.
5. Setelah instalasi selesai, R akan terinstal di komputer Anda.

Linux

Di sistem Linux, Anda dapat menggunakan perintah terminal untuk menginstal R. Berikut adalah contoh untuk beberapa distribusi umum:

Ubuntu/Debian:

Buka Program `csharp` anda dan run koding dibawah ini!

```
Copy code
sudo apt-get update
sudo apt-get install r-base
```

CentOS/Fedora:

Buka Program Command Prompt anda dan run koding dibawah ini!

```
sudo yum install R
```

1.4.2 Instalasi RStudio:

RStudio adalah Integrated Development Environment (IDE) yang mempermudah pengembangan dalam R. Ikuti langkah-langkah di bawah ini untuk menginstal RStudio:

Windows, macOS, dan Linux:

1. Kunjungi situs resmi RStudio di <https://www.rstudio.com/products/rstudio/download/>.

2. Pilih “RStudio Desktop” yang sesuai dengan sistem operasi Anda.
3. Unduh installer RStudio dan jalankan file installer yang diunduh.
4. Ikuti panduan instalasi dan pilih opsi default kecuali jika Anda tahu persis apa yang Anda lakukan.
5. Setelah instalasi selesai, RStudio akan terinstal di komputer Anda.

1.5 Video Instalasi R & RStudio

1.5.1 Windows

1.5.2 MacOS

1.6 Interface R & RStudio:

Interface adalah tampilan aplikasi R dan Rstudio yang telah terpasang diperlihatkan pada Gambar 1.1 dan Gambar 1.2.

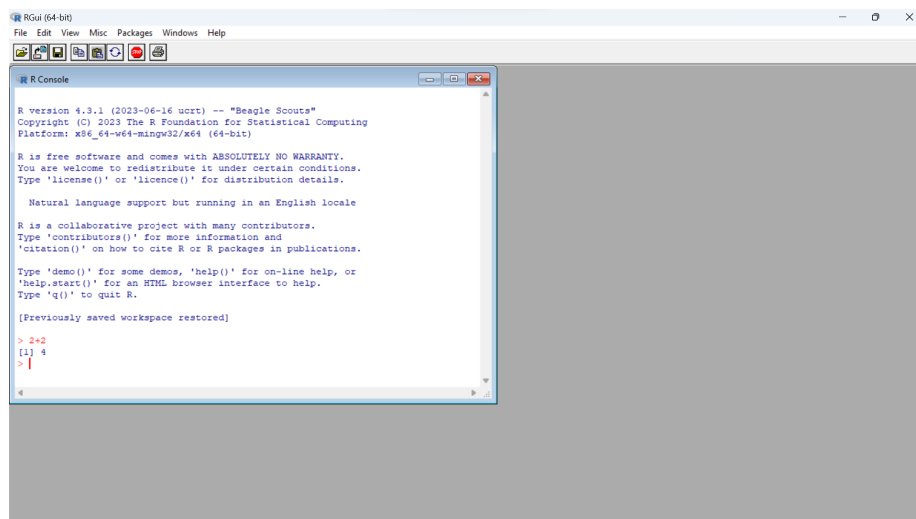


Figure 1.1: Jendela R.

Tampilan ini memiliki beberapa komponen utama, termasuk:

- **Script Panel:** Tempat Anda menulis kode R dalam skrip.
- **Console Panel:** Tempat hasil dari kode R ditampilkan, serta tempat Anda dapat menjalankan kode secara interaktif.
- **Environment Panel:** Menampilkan daftar variabel yang ada dalam sesi R Anda.

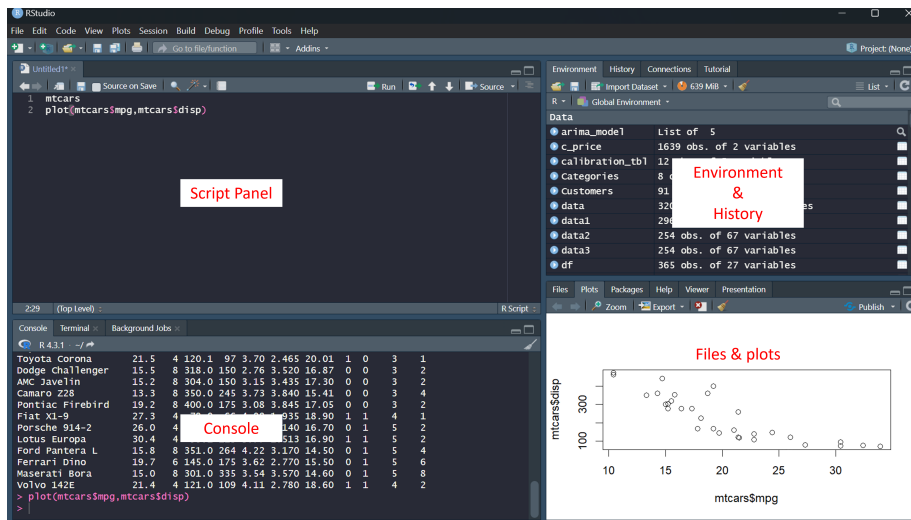


Figure 1.2: Jendela RStudio.

- **History Panel:** Menampilkan riwayat perintah yang telah dijalankan.
- **Files/Plots/Packages/Help Panel:** Panel tambahan yang membantu Anda mengelola file, visualisasi, pustaka, dan panduan bantuan.

Interface ini memudahkan pengguna untuk menulis, menjalankan, dan mengelola kode R serta menganalisis data dengan nyaman.

1.7 Sintaks Dasar R

Berikut ini beberapa kode sederhana yang bisa dipelajari untuk memulai memahami cara kerja Bahasa pemrograman R.

```
3+7
3-7
3^7
3/7
3*7
9^(1/3)
```

1.8 Bantuan (Help) R

Salah satu bagian terpenting dalam bekerja dengan bahasa R adalah mengetahui di mana mencari bantuan. R memiliki beberapa fasilitas in-line, selain berbagai

sumber daya bantuan di ekosistem R. Anda dapat menggunakan bantuan untuk fungsi tertentu.

```
help.start()      # menu di mana Anda dapat menavigasi bantuan lokal berbasis web
?help            # menu di mana Anda dapat menavigasi bantuan lokal berbasis web
?class           # mendapatkan bantuan untuk fungsi `class`
help(class)      # mendapatkan bantuan untuk fungsi `class`
??class          # jika Anda tidak tahu nama fungsi yang Anda cari
help.search('class') # jika Anda tidak tahu nama fungsi yang Anda cari
```

1.9 Shortcut Penggunaan Rstudio

Beberapa petunjuk bermanfaat untuk Rstudio (IDE) meliputi:

Kata Kunci	Perintah	Detail
Ctrl + Return (Enter)	untuk menjalankan baris dari editor	~
Ctrl + Shift + #	untuk fokus pada tab bantuan	kontradiktif
Alt + Shift + k	untuk jalur pintas keyboard RStudio	~
Ctrl + r	untuk menelusuri sejarah perintah	~
Alt + Shift + j	untuk menavigasi antar bagian kode	~
Ctrl + 1	untuk melompat ke editor	tab untuk penyelesaian otomatis
Ctrl + 2	untuk melompat ke konsol	tab untuk penyelesaian otomatis
Ctrl + 8	untuk melompat ke environment list	tab untuk pelengkapan otomatis
Alt + l	Collapse chunk	Code Folding
Alt + Shift + l	Unfold chunk	Code Folding
Alt + o	Collapse all	Code Folding
Alt + Shift + o	Unfold all	Code Folding
Alt + “-”	untuk operator penugasan <-	~
Alt + Shift + c	kode komentar/tanda komentar dalam file	.R kontradiktif

1.10 Praktikum

Buatlah tutorial Instalasi R dan R Studio dalam M.word! Lengkapi setiap prosesnya dengan gambar dan penjelasan.

Chapter 2

Dasar Pemrograman R

Pemrograman R merujuk pada proses menulis kode dan mengembangkan program menggunakan bahasa pemrograman R. R adalah bahasa pemrograman yang fokus pada analisis statistik, manipulasi data, dan visualisasi. Pada bab ini akan dibahas beberapa unsur utama dalam pemrograman menggunakan bahasa pemrograman R.

2.1 Variabel

Variabel dalam bahasa pemrograman R digunakan untuk menyimpan dan mengelola data. Variabel memungkinkan Anda untuk menampung nilai-nilai berbagai jenis, seperti angka, karakter (teks), atau nilai logika (benar/salah). Berikut ini adalah cara untuk mendefinisikan dan menggunakan variabel dalam R:

2.1.1 Mendefinisikan Variabel

Untuk membuat variabel, Anda cukup menggunakan tanda `<-` atau `=` untuk memberikan nilai pada variabel.

```
x <- 10      # Mendefinisikan variabel x
y = 12      # Mendefinisikan variabel y
```

2.1.2 Aturan Nama Variabel

- Nama variabel harus dimulai dengan huruf atau tanda `.`
- Nama variabel bisa terdiri dari huruf, angka, dan tanda `_`.

- Karakter khusus seperti $+$, $-$, $*$, $/$, $^$ dll. tidak diperbolehkan dalam nama variabel.
- Nama variabel bersifat case-sensitive, artinya x dan X dianggap berbeda.

2.2 Operator

Operator adalah simbol yang mengarahkan compiler untuk melakukan berbagai macam operasi terhadap beberapa penugasan. Operator mensimulasikan berbagai operasi matematis, logika, dan keputusan yang dilakukan pada sekumpulan Bilangan Kompleks, Integer, dan Numerik sebagai penugasan masukan (input). R mendukung sebagian besar empat jenis operator biner antara satu set penugasan. Dalam ini, kita akan melihat berbagai jenis operator yang tersedia di R penggunaannya.

2.2.1 Aritmatika

Penggunaan operator aritmatika dalam program R adalah untuk mensimulasikan berbagai operasi matematika, seperti penambahan, pengurangan, perkalian, pembagian, dan modulo. Operator aritmatika yang dilakukan bisa saja berupa nilai skalar, bilangan kompleks, atau vektor.

Operator	R
Penjumlahan	$+$
Pengurangan	$-$
Perkalian	$*$
Divisi/Pembagian	$/$
Pemangkatan	$^$
Modulo	$\% \%$

Perhatikan cuplikan R berikut:

```
x <- c(2,3,5)    # memuat vektor x
y <- c(2,4,6)    # memuat vektor y
x+y              # hasil penjumlahan vektor x dan y
print (x+y)      # hasil penjumlahan vektor x dan y
print (x-y)      # hasil pengurangan vektor x dan y
print (x*y)      # hasil perkalian vektor x dan y
print (x/y)      # hasil pembagian vektor x dan y
print (x^y)      # hasil pemangkatan vektor x dan y
print (x%%y)     # hasil modulo vektor x dan y
```


Adakalanya anda perlu menampilkan keterangan/komentar yang juga melekat pada hasil perhitungan R itu sendiri. Maka anda dapat melakukannya dengan cara berikut:

```
cat("Penjumlahan vektor x dan y :", x + y, "\n")
cat("Pengurangan vektor x dan y :", x - y, "\n")
cat("Perkalian vektor x dan y :", x * y, "\n")
cat("Pembagian vektor x dan y :", x / y, "\n")
cat("Pemangkatan vektor x dan y :", x ^ y)
cat("Modulo vektor x dan y :", x %% y, "\n")
```

Catatan: Penjelasan lebih lengkap mengenai modulo dapat lihat pada link ini

2.2.2 Relasional

Operator relasional melakukan operasi perbandingan antara elemen yang bersesuaian pada setiap operan. Mengembalikan nilai Boolean TRUE jika operan pertama memenuhi relasi dibandingkan dengan operan kedua. Nilai TRUE selalu dianggap lebih besar dari FALSE.

Operator	R	Keterangan
Kurang dari	<	Mengembalikan TRUE jika elemen yang bersesuaian pada operan pertama lebih kecil dari operan kedua. Selain itu akan mengembalikan FALSE
Kurang dari sama dengan	<=	Mengembalikan TRUE jika elemen yang bersesuaian pada operan pertama kurang dari atau sama dengan elemen operan kedua. Selain itu akan mengembalikan FALSE
Lebih besar dari	>	Mengembalikan TRUE jika elemen yang bersesuaian pada operan pertama lebih besar dari operan kedua. Selain itu akan mengembalikan FALSE

Lebih besar dari sama dengan	<code>>=</code>	Mengembalikan BENAR jika elemen yang bersesuaian pada operan pertama lebih besar atau sama dengan dari operan kedua. Selain itu akan mengembalikan FALSE
Sama Dengan	<code>==</code>	Mengembalikan BENAR jika dan hanya jika kedua sisi bernilai sama
Tidak Sama dengan	<code>!=</code>	Mengembalikan BENAR jika elemen yang bersesuaian pada operan pertama tidak sama dengan dari operan kedua

```
x <- c(2,3,5)    # memuat vektor x
y <- c(2,4,6)    # memuat vektor y
cat("Vektor x kurang dari Vektor y:", x < y, "\n")
cat("Vector x kurang dari sama dengan Vector y:", x <= y, "\n")
cat("Vector x lebih besar dari Vector y :", x > y, "\n")
cat("Vector x lebih besar dari sama dengan Vector y :", x >= y, "\n")
cat("Vector x sama dengan Vector y:", x == y, "\n")
cat("Vector x tidak sama dengan Vector y:", x != y)
```

2.2.3 Logika

Operator logis mensimulasikan operasi keputusan, berdasarkan operator yang ditentukan antara operan, yang kemudian dievaluasi ke nilai Boolean Benar atau Salah. Nilai bilangan bulat bukan nol dianggap sebagai nilai BENAR, baik itu bilangan kompleks atau bilangan real.

Operator	R	Keterangan
NOT	<code>!</code>	Operasi negasi/kebalikan pada status elemen operan
AND	<code>&</code>	Mengembalikan TRUE jika kedua operan bernilai Benar
OR	<code> </code>	Mengembalikan TRUE jika salah satu operan adalah Benar

XOR	\wedge	Mengembalikan TRUE jika salah satu dari kedua elemen pertama operan bernilai Benar
-----	----------	--

```
x <- c(0,TRUE,FALSE)
y <- c(TRUE,0.1,4+3i)

# Melakukan operasi logika pada Operan
cat("Logika Negasi (~) untuk vektor x:", !x, "\n")
cat("Logika Negasi (~) untuk vektor y :", !y, "\n")
cat ("Logika Konjungsi (Dan) :", x & y, "\n")
cat ("Logika Disjungsi (Atau) :", x | y, "\n")
```

2.2.4 Operator Lain-lain

Berikut ini juga ada beberapa operator yang kemungkinan besar juga akan anda perlukan pada saat akan menggunakan R.

```
x <- c(2,3,5)           # memuat vektor x
y <- c(2,4,6)           # memuat vektor y
sqrt(x*y)               # Bentuk akar
log(x)                  # logaritma
exp(y)                  # eksponen
(x/y) + y               # Tanda kurung
```

Catatan: Sifat Komutatif Asosiatif dan Distributif juga berlaku dalam program R.

2.3 Tipe Data

Dalam pemrograman seperti R dan Python, tipe data merupakan konsep penting. Keduanya dapat menggunakan variabel untuk menyimpan tipe yang berbeda-beda, berikut adalah tipe data paling mendasar yang harus diketahui:

Tipe Data	R	Penjelasan
Double/Float	5.6	Bilangan yang mempunyai koma
Integer	5	Bilangan bulat 1,2,...,n

Boolean/Logical	TRUE/FALSE	Benar bernilai 1 dan Salah bernilai 0
String/Character	'Dsciencelabs'	karakter/kalimat bisa berupa huruf angka, dll (diapit tanda " atau ')
Complex	1 + 5i	Pasangan angka real dan imajiner

Berikut ini adalah koding R yang dapat digunakan untuk menetapkan kelima tipe data diatas:

```
d1 = 5.6           # Tetapkan nilai desimal
d2 = as.integer(5) # tetapkan nilai integer
d2 = 5L           # cara lain untuk memuat nilai integer di R
d3 = c(TRUE,FALSE) # Boolean/Logical
d3 = as.logical(c(0,1)) # cara lain untuk memuat Boolean/Logical
d4 = c("a","b",'123') # String/Character
d5 = 1 + 5i        # Complex
```

Untuk memeriksa tipe data dalam R:

```
class(d1)          # cetak nama kelas variabel
typeof(d1)         # cetak tipe variabel x
```

2.4 Statistika Dasar

```
data <- c(85, 90, 78, 92, 88) # Data
sum(data)                   # Jumlahan data
length(data)                # Banyaknya data
mean (data)                 # Menghitung rata-rata
var (data)                  # Menghitung variansi
sd (data)                   # Simpangan baku
min(data)                   # Minimum
max(data)                   # Maksimum
```

2.5 Contoh Pemrograman Dasar

Berikut ini dilampirkan contoh kasus dasar pemrograman R.

2.5.1 Menghitung Rata-rata

```
data <- c(85, 90, 78, 92, 88)
jumlah_data <- length(data)
total <- sum(data)
rata_rata <- total / jumlah_data
print(paste("Rata-rata:", rata_rata))
```

```
## [1] "Rata-rata: 86.6"
```

2.5.2 Membandingkan Angka

```
a <- 10
b <- 20

if (a < b) {
  print("a lebih kecil dari b")
} else if (a > b) {
  print("a lebih besar dari b")
} else {
  print("a sama dengan b")
}
```

```
## [1] "a lebih kecil dari b"
```

2.5.3 Membandingkan Karakter

```
kata1 <- "Apel"
kata2 <- "apel"

if (kata1 == kata2) {
  print("kata1 sama dengan kata2")
} else {
  print("kata1 berbeda dari kata2")
}
```

```
## [1] "kata1 berbeda dari kata2"
```

2.5.4 Mengecek Kondisi Gabungan

```
umur <- 25
pendapatan <- 5000

if (umur > 18 && pendapatan > 4000) {
  print("Anda memenuhi syarat")
} else {
  print("Anda tidak memenuhi syarat")
}
```

```
## [1] "Anda memenuhi syarat"
```

2.5.5 Penggunaan Operator Logika

```
x <- 5
y <- 10

if (x > 0 || y > 0) {
  print("Salah satu variabel positif")
} else {
  print("Kedua variabel non-positif")
}
```

```
## [1] "Salah satu variabel positif"
```

2.5.6 Pemeriksaan Kondisi dengan ifelse()

```
nilai <- 75
keterangan <- ifelse(nilai >= 70, "Lulus", "Tidak lulus")
print(paste("Nilai:", nilai, "--> Status:", keterangan))
```

```
## [1] "Nilai: 75 --> Status: Lulus"
```

2.6 Latihan

Berikut adalah beberapa contoh soal latihan dasar pemrograman dalam bahasa pemrograman R:

1. **Menghitung Luas Lingkaran:** Buatlah sebuah program R yang menerima input berupa jari-jari lingkaran dan menghitung serta mencetak luas lingkaran.
2. **Konversi Suhu:** Buatlah sebuah program R yang dapat mengonversi suhu dari Celsius ke Fahrenheit. Program harus menerima input suhu dalam Celsius dan menghasilkan output suhu dalam Fahrenheit.
3. **Menghitung Faktorial:** Buatlah sebuah program R yang menghitung faktorial dari sebuah bilangan bulat positif. Program harus menerima input bilangan bulat positif dan menghasilkan output faktorialnya.
4. **Mencari Bilangan Prima:** Buatlah sebuah program R yang menerima input sebuah bilangan bulat dan menghasilkan output apakah bilangan tersebut merupakan bilangan prima atau bukan.
5. **Menghitung Pangkat:** Buatlah sebuah program R yang menerima input bilangan dasar dan eksponen, kemudian menghitung hasil dari bilangan dasar dipangkatkan dengan eksponen tersebut.
6. **Menghitung Total Nilai:** Buatlah sebuah program R yang menerima input sejumlah nilai mata kuliah dan menghitung total nilai serta rata-ratanya. Program harus menerima input nilai-nilai mata kuliah dan menghasilkan total nilai serta rata-ratanya.
7. **Menentukan Ganjil/Genap:** Buatlah sebuah program R yang menerima input bilangan bulat dan mencetak apakah bilangan tersebut ganjil atau genap.
8. **Menghitung Keliling dan Luas Persegi:** Buatlah sebuah program R yang menerima input panjang sisi persegi dan menghitung serta mencetak keliling dan luasnya.

Chapter 3

Struktur Data

Struktur data dalam R adalah cara di mana Anda dapat mengatur dan menyimpan data dalam bentuk yang terstruktur agar mudah diakses, dikelola, dan dimanipulasi. Struktur data memungkinkan Anda untuk mengelompokkan nilai-nilai data ke dalam objek yang sesuai dengan jenis dan sifat data yang Anda miliki. R memiliki beberapa jenis struktur data yang dapat digunakan untuk berbagai tujuan.

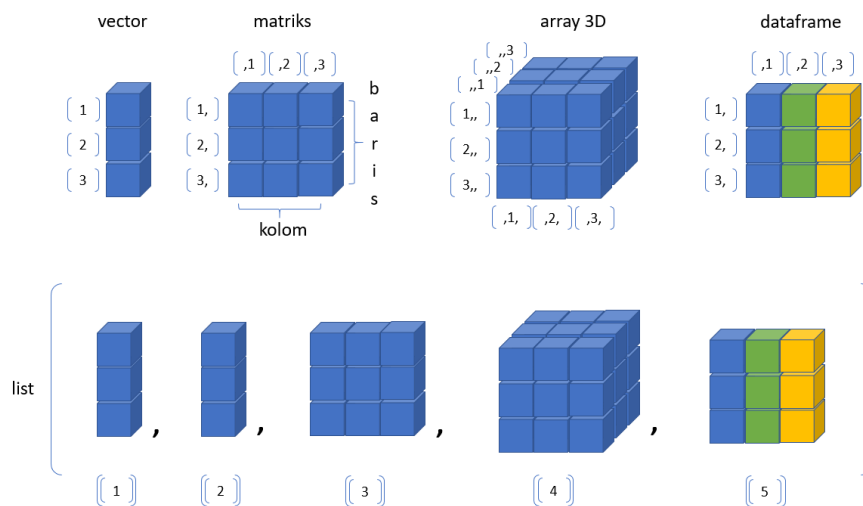


Figure 3.1: Struktur Data dalam R

3.1 Vektor

Elemen paling dasar dalam R adalah vektor, yang berisikan kumpulan elemen data dengan tipe yang sama. Terdapat dua jenis vector, yaitu vector numerik dan vector karakter. Misalnya:

```
vektor <- seq(from=10, to=21, by=1)      # fungsi `seq()` dengan "by"
vektor<- seq(from=10, to=21, len=12)     # fungsi `seq()` dengan "len"
vektor <- 10:21                          # tetapkan data dalam vektor
vektor <- vektor+2                        # Operasi berdasarkan elemen
vektor <- vektor*2                       # Tambahkan 2 untuk setiap elemen
vektor <- vektor^2                       # Pangkat 2 untuk setiap elemen
vektor <- sqrt(vektor)                   # Akar kuadrat untuk setiap elemen
vektor <- log(vektor)                    # Logaritma untuk setiap elemen
vektor<- c(0.5, 0.6)                     # Numerik
vektor <- c(TRUE, FALSE)                 # Logis
vektor <- c(T, F)                        # Logis
vektor <- c("a", "b", "c")               # Karakter
vektor <- 9:29                           # Integer
vektor <- c(1+0i, 2+4i)                   # Kompleks
vektor <- vector("numeric", length = 10) # untuk inisialisasi vektor.
```

Catatan: Menurut dokumentasi R untuk `typeof()` dan `class()`, pernyataan tentang “perbedaan utama/main difference” adalah tidak benar. Kelas adalah atribut dari objek yang dapat ditetapkan terlepas dari mode penyimpanan internalnya, sedangkan `typeof()` menentukan tipe (R internal) atau mode penyimpanan dari objek apa pun. Salah satu menggambarkan karakteristik logis sedangkan yang lain adalah karakteristik fisik dari suatu objek.

```
class(vektor)                          # Periksa kelas vektor
as.numeric(vektor)                      # Menetapkan vektor sebagai numerik
as.logical(vektor)                      # Menetapkan vektor sebagai logis
as.character(vektor)                    # Menetapkan vektor sebagai karakter
as.numeric(c(FALSE,TRUE,TRUE,FALSE))    # Menetapkan vektor logis sebagai angka
```

Terkadang, R tidak dapat menemukan cara untuk memaksa suatu objek dan ini dapat menghasilkan NA.

```
vektor <- c("a", "b", "c", "1")          # menetapkan nilai vektor
as.numeric(vektor)                       # menetapkan vektor sebagai numerik
```

```
## Warning: NAs introduced by coercion
```

```
as.logical(vektor)           # menetapkan vektor sebagai logis
as.complex(vektor)           # menetapkan vektor sebagai karakter
```

```
## Warning: NAs introduced by coercion
```

Catatan: Saat paksaan tidak masuk akal terjadi, Anda biasanya akan mendapat peringatan dari R.

Kita sudah melihat bahwa elemen dasar dari objek R adalah vektor. Vektor dapat ditetapkan dengan berbagai jenis berikut:

- **character:** di mana setiap elemen adalah string, mis., urutan simbol alfanumerik.
- **numeric:** di mana setiap elemen adalah bilangan real dalam format floating point presisi ganda.
- **integer:** di mana setiap elemen adalah integer.
- **logis:** di mana setiap elemen adalah TRUE, FALSE, atau NA3
- **complex:** di mana setiap elemen adalah bilangan kompleks.

3.2 Matriks

Matriks adalah vektor dengan atribut dimensi. Matriks dibuat berdasarkan kolom, sehingga entri dapat dianggap dimulai dari sudut “kiri atas” dan mengalir di kolom.

```
matriks <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
matriks
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

Matriks juga dapat dibuat langsung dari vektor dengan menambahkan atribut dimensi.

```
matriks <- 1:6           # Membuat vektor
dim(matriks) <- c(2, 3)  # rubah vektor sebagai matriks sebesar 2x3
matriks                  # Mencetak hasilnya
```

Matriks dapat dibuat dengan pengikatan kolom atau pengikatan baris dengan fungsi `cbind()` dan `rbind()`.

```
x <- 1:3 # Membuat vektor `x`
y <- 10:12 # Membuat vektor `y`
cbind(x, y) # Menggabungkan vektor `x` dan `y` dengan kolom
rbind(x, y) # Menggabungkan vektor `x` dan `y` dengan baris
```

3.3 Array

Array mirip dengan matrix, tetapi dapat memiliki lebih dari dua dimensi. Masing-masing dimensi dalam array memiliki ukuran tertentu.

```
array_data <- array(c(1, 2, 3, 4, 5, 6), dim = c(2, 3, 1))
```

3.4 Faktor

Faktor-faktor digunakan untuk mewakili data kategorikal dan dapat menjadi tidak teratur atau teratur. Orang dapat menganggap faktor sebagai vektor integer di mana setiap integer memiliki label. Menggunakan faktor dengan label lebih baik daripada menggunakan bilangan bulat karena faktor menggambarkan diri sendiri. Memiliki variabel yang memiliki nilai “Laki-laki” dan “Perempuan” lebih baik daripada variabel yang memiliki nilai 1 dan 2. Objek-objek dapat dibuat dengan fungsi `faktor()`.

```
x <- faktor(c("yes", "no", "yes", "no")) # Membuat objek faktor
x # Cetak hasilnya
table(x) # Tabel dari `x`
unclass(x) # Melihat representasi faktor yang mendasarinya
attr(x, "levels") # Melihat representasi faktor yang mendasarinya
```

3.5 Data Frame

Kerangka data (data frame) adalah tabel atau struktur mirip array dua dimensi di mana setiap kolom berisi nilai satu variabel dan setiap baris berisi satu set nilai dari setiap kolom.

Berikut ini adalah karakteristik data frame.

- Nama kolom tidak boleh kosong;
- Nama baris harus unik;
- Data yang disimpan dalam data frame bisa dari numerik, faktor atau tipe karakter;

- Setiap kolom harus berisi jumlah item data yang sama.

```
# Buat data frame pertama.
df1 <- data.frame(id = c (1:5),
                  name = c("Julian","Vanessa","Jeffry","Angel","Nikki"),
                  salary = c(623.3,515.2,611.0,729.0,843.25),
                  start_date = as.Date(c("2022-01-01", "2022-09-23", "2022-11-15",
                  dept = c("DS","DS","BA","DA","DS"), stringsAsFactors = F)
df1

# Buat data frame kedua.
df2 <-data.frame(id = c (6:10),
                 name = c("Ardifo","Irene","Kefas","Sherly","Bakti"),
                 salary = c(578.0,722.5,632.8,632.8,NA),
                 start_date = as.Date(c("2022-05-21","2022-07-30","2022-06-17",
                 dept = c("Actuaries","Actuaries","CA","DE","Lecturer"),stringsAsFactors = F)
df2

df3 <- rbind(df1,df2)           # Gabungkan dua frame data
print(df3)                     # Cetak hasilnya `df3`
head(df3)                      # Cetak enam baris pertama
head(df3,6)                    # Cetak enam baris pertama
#View(df3)                     # Menggunakan RStudio seperti penampil Excel
class(df3)                     # objeknya bertipe data.frame
str(df3)                       # Dapatkan struktur data frame
dim(df3)                       # Periksa dimensi data
```

Data frame biasanya dibuat dengan membaca dalam dataset menggunakan `read.table()` atau `read.csv()`. Namun, data frame juga dapat dibuat secara eksplisit dengan fungsi `data.frame()` atau mereka dapat dipaksakan dari jenis objek lain seperti list.

3.6 Lists

List dalam R adalah struktur data yang mengizinkan Anda untuk menyimpan berbagai jenis objek, termasuk vektor, matriks, array, dataframe, dan objek list lainnya, dalam satu objek tunggal. Ini memungkinkan Anda untuk membuat struktur data yang kompleks dan fleksibel dengan menggabungkan objek-objek yang berbeda ke dalam satu wadah. List sering digunakan ketika Anda perlu mengorganisir dan mengelompokkan objek-objek yang terkait.

Berikut adalah contoh penggunaan dan pembuatan list dalam R:

```
# Membuat vektor dan matriks
vektor <- c(1.5, 2.7, 3.2, 4.0)
matriks <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
```

```
# Membuat dataframe
data_frame <- data.frame(name = c("Alice", "Bakti", "Charlie"),
                          age = c(25, 30, 28),
                          score = c(95, 88, 76))
```

```
faktor <- "List, Sudah Jadi"
```

```
# Membuat list
my_list <- list(vektor, matriks, data_frame, faktor)
```

```
# Menampilkan list
print(my_list)
```

Anda juga dapat memberi nama pada setiap elemen dalam list untuk membuat list yang lebih mudah dibaca:

```
nama_list <- list(elemen1 = vektor,
                  elemen2 = matriks,
                  elemen3 = data_frame,
                  elemen4 = faktor)

# Menampilkan elemen dalam list berdasarkan nama
print(nama_list$elemen1)
```

```
## [1] 1.5 2.7 3.2 4.0
```

```
print(nama_list$elemen2)
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
print(nama_list$elemen3)
```

```
##      name age score
## 1  Alice  25    95
## 2  Bakti  30    88
## 3 Charlie  28    76
```

```
print(nama_list$elemen4)
```

```
## [1] "List, Sudah Jadi"
```

Anda dapat mengakses elemen-elemen dalam list menggunakan indeks atau nama. Misalnya:

```
# Mengakses elemen pertama dalam list menggunakan indeks
elemen1 <- my_list[[1]]
elemen2 <- my_list[[2]]
elemen3 <- my_list[[2]]
elemen4 <- my_list[[2]]

# Menampilkan hasil
print(elemen1)
print(elemen2)
print(elemen3)
print(elemen4)
```

List memungkinkan Anda mengorganisir, mengelompokkan, dan mengakses objek-objek yang beragam dalam struktur data tunggal, sehingga sangat berguna dalam analisis data yang kompleks dan beragam.

3.7 Rekayasa Data Frame

3.7.1 Tanpa Packages

Sebagai seorang Data Scientist, ketika mencoba menyimulasikan proses analisis data, pemodelan, bahkan prediksi. Anda harus mampu secara intuitif membangun dataframe untuk memperkirakan kumpulan data sampel. Terutama, ketika Anda tidak memiliki kumpulan data sampel sama sekali. Oleh karena itu, pada bagian ini, kita akan belajar sedikit mengenai cara menghasilkan dataframe. Harap perhatikan baik-baik contoh berikut:

```
# Misalkan Anda ingin membangun kumpulan data karyawan di sebuah perusahaan

No<-(1:52) # Menghasilkan bilangan 1-52
Name<-c(LETTERS,letters) # 26 LETTERS dan 26 letters
Gender<-sample(rep(c("Male","Female"),times=26)) # 26 Laki-laki dan 26 perempuan

# Menghasilkan tanggal lahir
year_in_3<-seq(as.Date("2000/01/01"), by="year", length.out=4)
```

```

Birthday <- rep(year_in_3, times=13)

# Menghasilkan kategori universitas
univ1<-rep("National",times=26)      # 26 universitas negeri
univ2<-rep("Private",times=16)       # 16 universitas swasta
univ3<-rep("Overseas",times=10)      # 10 universitas luar negeri
Universities<-sample(c(univ1,univ2,univ3)) # Menggabungkan data (vetor)

gpa<-runif(52,min=3.00,max=4.00)      # Menghasilkan 52 bilangan acak (min=
GPA<-round(gpa,digits=2)              # Mengatur digit bilangan acak Anda
Salary<-sample(600:1200,52,replace=T) # Menghasilkn sampel antara 600-1200

Employees<-data.frame(No,
                        Name,
                        Birthday,
                        Gender,
                        Universities,
                        GPA,
                        Salary)

Employees

```

##	No	Name	Birthday	Gender	Universities	GPA	Salary
## 1	1	A	2000-01-01	Female	Private	3.97	1012
## 2	2	B	2001-01-01	Male	Private	3.07	1016
## 3	3	C	2002-01-01	Female	Overseas	3.69	985
## 4	4	D	2003-01-01	Male	Private	3.28	981
## 5	5	E	2000-01-01	Male	Private	3.53	720
## 6	6	F	2001-01-01	Female	National	3.30	1122
## 7	7	G	2002-01-01	Male	Private	3.58	754
## 8	8	H	2003-01-01	Male	Private	3.38	999
## 9	9	I	2000-01-01	Female	National	3.47	1200
## 10	10	J	2001-01-01	Female	National	3.78	886
## 11	11	K	2002-01-01	Female	Overseas	3.25	891
## 12	12	L	2003-01-01	Male	Overseas	3.53	809
## 13	13	M	2000-01-01	Male	National	3.72	605
## 14	14	N	2001-01-01	Female	National	3.92	977
## 15	15	O	2002-01-01	Female	Overseas	3.83	822
## 16	16	P	2003-01-01	Female	Overseas	3.59	930
## 17	17	Q	2000-01-01	Male	National	3.08	672
## 18	18	R	2001-01-01	Female	Private	3.22	1139
## 19	19	S	2002-01-01	Male	National	3.11	990
## 20	20	T	2003-01-01	Male	Private	3.76	913
## 21	21	U	2000-01-01	Female	National	3.45	754
## 22	22	V	2001-01-01	Male	National	3.86	1166
## 23	23	W	2002-01-01	Male	Private	3.58	1128
## 24	24	X	2003-01-01	Female	National	3.92	971

##	25	25	Y	2000-01-01	Male	Overseas	3.66	841
##	26	26	Z	2001-01-01	Male	Private	3.86	825
##	27	27	a	2002-01-01	Male	National	3.30	815
##	28	28	b	2003-01-01	Female	National	3.08	956
##	29	29	c	2000-01-01	Male	National	3.16	815
##	30	30	d	2001-01-01	Male	National	3.02	773
##	31	31	e	2002-01-01	Female	National	3.07	1022
##	32	32	f	2003-01-01	Female	Private	3.57	708
##	33	33	g	2000-01-01	Female	National	3.42	971
##	34	34	h	2001-01-01	Male	National	3.23	992
##	35	35	i	2002-01-01	Female	Private	3.22	1014
##	36	36	j	2003-01-01	Female	Private	3.24	775
##	37	37	k	2000-01-01	Female	National	3.43	1071
##	38	38	l	2001-01-01	Male	Private	3.29	1062
##	39	39	m	2002-01-01	Male	Private	3.59	799
##	40	40	n	2003-01-01	Male	Overseas	3.86	989
##	41	41	o	2000-01-01	Male	National	3.44	730
##	42	42	p	2001-01-01	Female	National	3.92	972
##	43	43	q	2002-01-01	Female	National	3.59	668
##	44	44	r	2003-01-01	Male	National	3.32	1073
##	45	45	s	2000-01-01	Female	Private	3.18	1061
##	46	46	t	2001-01-01	Male	National	3.67	754
##	47	47	u	2002-01-01	Female	National	3.42	1066
##	48	48	v	2003-01-01	Female	Overseas	3.10	855
##	49	49	w	2000-01-01	Female	Overseas	3.40	1098
##	50	50	x	2001-01-01	Female	National	3.26	693
##	51	51	y	2002-01-01	Male	National	3.28	607
##	52	52	z	2003-01-01	Male	Overseas	3.10	634

3.7.2 Menggunakan Packages

Dalam contoh kedua ini, digunakan pustaka (*Packages*) *faker* untuk menghasilkan data palsu seperti nama, alamat, dan lain-lain. Pastikan Anda telah menginstal pustaka tersebut menggunakan perintah `install.packages("faker")` jika belum terinstal, mengikuti langkah berikut.

```
install.packages("remotes")
remotes::install_github("ThinkR-open/faker")
```

Selanjutnya, anda dapat membuat data frame palsu seperti diperlihatkan berikut:

```
library(faker)
fake_ticket_client(vol = 10)
```

```
## # A tibble: 10 x 25
##   ref      num_client first last  job      age region
##   <chr>      <chr>      <chr> <chr> <chr> <dbl> <chr>
## 1 DOSS-AMQN~ 79      Jovan O'Ke~ Gene~ 22 Prove~
## 2 DOSS-NCKJ~ 69      Miss Lean~ Emer~ 68 Haute~
## 3 DOSS-GPBE~ 120     Odell Stok~ Engi~ 24 Auver~
## 4 DOSS-GRLN~ 31      Loren Lars~ <NA> NA Midi~
## 5 DOSS-LEPJ~ 59      Mayb~ Maye~ Furt~ 18 Aquit~
## 6 DOSS-DUCL~ 118     Jama~ Ober~ Engi~ 18 Île-d~
## 7 DOSS-OCED~ 77      Lee  Scha~ Admi~ NA Poito~
## 8 DOSS-KXSJ~ 65      Deme~ Auer  Cont~ 21 Midi~
## 9 DOSS-UITD~ 141     Wilf~ Harv~ Educ~ 53 <NA>
## 10 DOSS-SHKL~ 182     Addy~ Nien~ Earl~ 65 Rhône~
## # i 18 more variables: id_dpt <chr>,
## #   departement <chr>, cb_provider <chr>, name <chr>,
## #   entry_date <dtm>, fidelity_points <dbl>,
## #   priority_encoded <dbl>, priority <fct>,
## #   timestamp <date>, year <dbl>, month <dbl>,
## #   day <int>, supported <chr>,
## #   supported_encoded <int>, type <chr>, ...
```

Catatan: Pustaka *fakir* Menyimpan beberapa dataset didalamnya, antara lain:

```
fake_products(10) # Rekayasa Data Produk
fake_visits(from = "2017-01-01", to = "2017-01-31") # Pengunjung Website
fake_sondage_answers(n = 10) # Kuisioner transfortasi
```

3.8 Latihan

1. Buatlah Rekayasa dataframe Mahasiswa dengan empat kolom: “Nama”, “Usia”, “Kota”, dan “Nilai”. Sebanyak 100 baris, dengan syarat tidak boleh ada nama yang sama.
2. Buatlah Rekayasa dataframe Karyawan dengan tujuh kolom: “No”, “Name”, “Birthday”, “Gender”, “Universities”, “GPA”, “Salary”. Sebanyak 100 baris, dengan syarat tidak boleh ada nama yang sama.
3. Buatlah Rekayasa dataframe pengunjung Website, sebanyak 200 baris.

Catatan: Kumpulkan hasil latihan anda, tidak boleh sama dengan teman mahasiswa lainnya.

Chapter 4

Referensi

Berikut adalah beberapa referensi yang dapat Anda gunakan untuk mempelajari dasar-dasar pemrograman dalam bahasa R:

1. Venables, W.N. Smith D.M. and R Core Team. 2018. **An Introduction to R**: <https://cran.r-project.org/manuals.html>
2. R for Data Science: <https://r4ds.had.co.nz/>
3. Codecademy - Learn R : <https://www.codecademy.com/learn/learn-r>
4. DataCamp: <https://www.datacamp.com/courses/tech:r>
5. Primartha, R. 2018. **Belajar Machine Learning Teori dan Praktik**. Penerbit Informatika : Bandung
6. Rosadi,D. 2016. **Analisis Statistika dengan R**. Gadjah Mada University Press: Yogyakarta
7. STHDA. Running RStudio and Setting Up Your Working Directory - Easy R Programming .<http://www.sthda.com/english/wiki/running-rstudio-and-setting-up-your-working-directory-easy-r-programming#set-your-working-directory>
8. STDHA. **Getting Help With Functions In R Programming**. <http://www.sthda.com/english/wiki/getting-help-with-functions-in-r-programming> .